

MACHINE SR_M3

REFINES SR_M2

SEES SR3_Signal

VARIABLES

Route_Req
Route_Cel
Route_Occ
Route2Path
Block2Route
Route2OccPath
Point2Pos
Point2Mode
Route2Signal

INVARIANTS

inv1: $Route2Signal \in ROUTE \rightarrow SIGNAL$

inv2: $\forall r. r \in dom(Route2OccPath) \wedge Route2OccPath(r) \neq NullPath \Rightarrow Route2Signal(r) = Red$

EVENTS

Initialisation

begin

act1: $Route_Req := \emptyset$
act2: $Route_Cel := \emptyset$
act3: $Route_Occ := \emptyset$
act4: $Route2Path := \emptyset$
act5: $Block2Route := \emptyset$
act6: $Route2OccPath := \emptyset$
act7: $Point2Pos := Point2InitPos$
act8: $Point2Mode := POINT \times \{Unlock\}$
act9: $Route2Signal := ROUTE \times \{Red\}$

end

Event ATS_Request $\langle ordinary \rangle \hat{=}$

extends ATS_Request

any

r

where

grd1: $r \notin Route_Req$

then

act1: $Route_Req := Route_Req \cup \{r\}$

end

Event ATS_Cancel $\langle ordinary \rangle \hat{=}$

extends ATS_Cancel

any

r

where

grd1: $r \in Route_Req$

then

act1: $Route_Cel := Route_Cel \cup \{r\}$

end

Event Route_Reserve $\langle ordinary \rangle \hat{=}$

extends Route_Reserve

any

r

where

grd1: $r \in Route_Req$

grd2: $r \notin Route_Cel$

grd3: $r \notin dom(Route2Path)$

grd4: $PathConflict[Route2InitPath[\{r\}]] \cap ran(Route2Path) = \emptyset$

```

    grd5:  $Block2Route^{-1}[\{r\}] = Path2Block[\{Route2InitPath(r)\}]$ 
    grd6:  $\forall po \cdot (po \in (Path2Block[Route2InitPath[\{r\}]] \cap POINT)) \Rightarrow (Point2Pos(po) = Route2Point2Pos(\{r \mapsto po\}))$ 
    grd7:  $\forall po \cdot (po \in (Path2Block[Route2InitPath[\{r\}]] \cap POINT)) \Rightarrow (Point2Mode(po) = Lock)$ 
  then
    act1:  $Route2Path := Route2Path \cup \{r \mapsto Route2InitPath(r)\}$ 
  end
Event Route_Cancel <ordinary>  $\hat{=}$ 
extends Route_Cancel
  any
     $r$ 
  where
    grd1:  $r \in Route\_Req$ 
    grd2:  $r \in Route\_Cel$ 
    grd3:  $r \notin Route\_Occ$ 
    grd4:  $Block2Route^{-1}[\{r\}] = \emptyset$ 
  then
    act1:  $Route\_Req := Route\_Req \setminus \{r\}$ 
    act2:  $Route\_Cel := Route\_Cel \setminus \{r\}$ 
    act3:  $Route2Path := \{r\} \triangleleft Route2Path$ 
  end
Event Route_Release <ordinary>  $\hat{=}$ 
extends Route_Release
  any
     $r$ 
  where
    grd1:  $r \in dom(Route2Path)$ 
    grd2:  $Route2Path(r) = NullPath$ 
    grd3:  $r \notin Route\_Occ$ 
    grd4:  $Block2Route^{-1}[\{r\}] = \emptyset$ 
  then
    act1:  $Route2Path := \{r\} \triangleleft Route2Path$ 
    act2:  $Route\_Req := Route\_Req \setminus \{r\}$ 
  end
Event Train_Enter <ordinary>  $\hat{=}$ 
extends Train_Enter
  any
     $r$ 
  where
    grd1:  $r \in dom(Route2Path)$ 
    grd2:  $r \notin Route\_Occ$ 
    grd3:  $r \notin dom(Route2OccPath)$ 
    grd4:  $Route2Signal(r) = Green$ 
  then
    act1:  $Route\_Occ := Route\_Occ \cup \{r\}$ 
    act2:  $Route2OccPath := Route2OccPath \cup \{r \mapsto NullPath\}$ 
  end
Event Train_Head_Move <ordinary>  $\hat{=}$ 
extends Train_Head_Move
  any
     $r$ 
     $op$ 
     $b$ 
  where
    grd1:  $r \in Route\_Occ$ 
    grd2:  $r \in dom(Route2Path)$ 
    grd3:  $b \in Path2Block[\{Route2Path(r)\}]$ 
    grd4:  $r \in dom(Route2OccPath)$ 

```

```

    grd5:  $op = Route2OccPath(r)$ 
    grd6:  $b \notin Path2Block[\{op\}]$ 
    grd8:  $Route2Signal(r) = Red$ 
  then
    act1:  $Route2OccPath(r) := PathIncrease(op)(b)$ 
  end
Event Train_Rear_Move  $\langle ordinary \rangle \hat{=}$ 
extends Train_Rear_Move
  any
     $r$ 
     $p$ 
     $sp$ 
     $op$ 
     $b$ 
  where
    grd1:  $r \in Route\_Occ$ 
    grd2:  $r \in dom(Route2Path)$ 
    grd3:  $p = Route2Path(r)$ 
    grd4:  $p \neq NullPath$ 
    grd5:  $sp \in PathSub[\{p\}]$ 
    grd6:  $r \in dom(Route2OccPath)$ 
    grd7:  $op = Route2OccPath(r)$ 
    grd8:  $b \in Path2Block[\{op\}] \cap Path2Block[\{p\}]$ 
    grd9:  $op \neq NullPath$ 
    grd10:  $sp = PathReduce(p)(b)$ 
    grd11:  $PathReduce(op)(b) = NullPath \Rightarrow card(Path2Block[\{p\}]) \neq 1$ 
  then
    act1:  $Route2Path(r) := sp$ 
    act2:  $Route2OccPath(r) := PathReduce(op)(b)$ 
  end
Event Train_Leave  $\langle ordinary \rangle \hat{=}$ 
extends Train_Leave
  any
     $r$ 
  where
    grd1:  $r \in Route\_Occ$ 
    grd2:  $r \in dom(Route2Path)$ 
    grd3:  $Route2Path(r) = NullPath$ 
    grd5:  $Route2OccPath(r) = NullPath$ 
  then
    act1:  $Route\_Occ := Route\_Occ \setminus \{r\}$ 
    act2:  $Route2OccPath := \{r\} \triangleleft Route2OccPath$ 
  end
Event Block_Reserve  $\langle ordinary \rangle \hat{=}$ 
extends Block_Reserve
  any
     $r$ 
     $p$ 
  where
    grd1:  $r \in Route\_Req$ 
    grd2:  $r \notin Route\_Cel$ 
    grd3:  $r \notin dom(Route2Path)$ 
    grd4:  $p = Route2InitPath(r)$ 
    grd5:  $Path2Block[\{p\}] \cap dom(Block2Route) = \emptyset$ 
  then
    act1:  $Block2Route := Block2Route \cup (Path2Block[\{p\}] \times \{r\})$ 
  end
Event Block_Cancel  $\langle ordinary \rangle \hat{=}$ 

```

```

extends Block_Cancel
  any
    r
  where
    grd1:  $r \in \text{Route\_Cel}$ 
    grd2:  $r \in \text{ran}(\text{Block2Route})$ 
    grd3:  $r \notin \text{dom}(\text{Route2OccPath})$ 
    grd4:  $\forall po \cdot po \in \text{Block2Route}^{-1}[\{r\}] \cap \text{POINT} \Rightarrow \text{Point2Mode}(po) = \text{Unlock}$ 
    grd5:  $\text{Route2Signal}(r) = \text{Red}$ 
  then
    act1:  $\text{Block2Route} := \text{Block2Route} \triangleright \{r\}$ 
  end
Event Block_Release  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Block_Release
  any
    b
    r
  where
    grd1:  $r \in \text{Route\_Occ}$ 
    grd2:  $b \in \text{Block2Route}^{-1}[\{r\}]$ 
    grd3:  $b \notin \text{Path2Block}[\{\text{Route2OccPath}(r)\}]$ 
    grd4:  $\langle \text{theorem} \rangle b \notin \text{Path2Block}[\{\text{Route2Path}(r)\}]$ 
    grd5:  $b \in \text{POINT} \Rightarrow \text{Point2Mode}(b) = \text{Unlock}$ 
    grd6:  $\text{Route2Signal}(r) = \text{Red}$ 
  then
    act1:  $\text{Block2Route} := \{b\} \triangleleft \text{Block2Route}$ 
  end
Event Point_Switch  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Point_Switch
  any
    po
    r
  where
    grd1:  $r \notin \text{Route\_Cel}$ 
    grd2:  $r \notin \text{dom}(\text{Route2Path})$ 
    grd3:  $\langle \text{theorem} \rangle po \in \text{Block2Route}^{-1}[\{r\}] \cap \text{POINT}$ 
    grd4:  $\text{Point2Mode}(po) = \text{Unlock}$ 
    grd5:  $\text{Point2Pos}(po) \neq \text{Route2Point2Pos}(\{r \mapsto po\})$ 
    grd6:  $\langle \text{theorem} \rangle \forall p \cdot p \in \text{ran}(\text{Route2OccPath}) \Rightarrow po \notin \text{Path2Block}[\{p\}]$ 
  then
    act1:  $\text{Point2Pos}(po) := \text{Route2Point2Pos}(\{r \mapsto po\})$ 
  end
Event Point_Lock  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Point_Lock
  any
    r
    po
  where
    grd1:  $r \notin \text{Route\_Cel}$ 
    grd2:  $r \notin \text{dom}(\text{Route2Path})$ 
    grd3:  $po \in \text{Block2Route}^{-1}[\{r\}] \cap \text{POINT}$ 
    grd4:  $\text{Point2Mode}(po) = \text{Unlock}$ 
    grd5:  $\text{Point2Pos}(po) = \text{Route2Point2Pos}(\{r \mapsto po\})$ 
  then
    act1:  $\text{Point2Mode} := \{po \mapsto \text{Lock}\} \triangleleft \text{Point2Mode}$ 
  end
Event Point_Unlock_Cancel  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Point_Unlock_Cancel

```

```

any
  r
  po
where
  grd1: r ∈ Route_Cel
  grd2: po ∈ Block2Route-1[{r}] ∩ POINT
  grd3: Point2Mode(po) = Lock
  grd4: r ∉ Route_Occ
  grd5: Route2Signal(r) = Red
then
  act1: Point2Mode := {po ↦ Unlock} ⇐ Point2Mode
end
Event Point_Unlock_Release ⟨ordinary⟩ ≐
extends Point_Unlock_Release
any
  r
  po
where
  grd1: r ∈ Route_Occ
  grd2: po ∈ POINT
  grd3: po ∈ Block2Route-1[{r}]
  grd4: po ∉ Path2Block[{Route2Path(r)}]
  grd6: Route2Signal(r) = Red
then
  act1: Point2Mode(po) := Unlock
end
Event Signal_Green_Reserve ⟨ordinary⟩ ≐
any
  r
where
  grd1: r ∈ dom(Route2Path)
  grd2: r ∈ Route_Req
  grd3: r ∉ Route_Cel
  grd4: Block2Route-1[{r}] = Path2Block[{Route2InitPath(r)}]
  grd5: ∀po.(po ∈ (Path2Block[Route2InitPath[{r}]] ∩ POINT)) ⇒ (Point2Pos(po) = Route2Point2Pos({r ↦ po}))
  grd6: ∀po.(po ∈ (Path2Block[Route2InitPath[{r}]] ∩ POINT)) ⇒ (Point2Mode(po) = Lock)
  grd7: r ∉ Route_Occ
  grd8: r ∉ dom(Route2OccPath)
then
  act1: Route2Signal := {r ↦ Green} ⇐ Route2Signal
end
Event Signal_Red_Cancel ⟨ordinary⟩ ≐
any
  r
where
  grd1: r ∈ Route_Cel
  grd2: Route2Signal(r) = Green
  grd3: r ∈ dom(Route2Path)
  grd4: r ∉ Route_Occ
then
  act1: Route2Signal := {r ↦ Red} ⇐ Route2Signal
end
Event Signal_Red_Occupied ⟨ordinary⟩ ≐
any
  r
where
  grd1: r ∈ dom(Route2Path)

```

```
    grd2:  $r \notin \text{Route\_Occ}$   
    grd3:  $\text{Route2Signal}(r) = \text{Green}$   
    grd4:  $r \in \text{dom}(\text{Route2OccPath})$   
    grd5:  $\text{Route2OccPath}(r) = \text{NullPath}$   
  then  
    act1:  $\text{Route2Signal}(r) := \text{Red}$   
  end  
END
```